

数据结构算法体系结构

复杂度

- 时间复杂度
  - $O(1)$  常数复杂度, 最佳, 比如 Hash 表、缓存等
  - $O(\log n)$  对数复杂度, 仅次于常数复杂度, 如二分查找、二叉搜索树等
  - $O(n)$  线性复杂度, 如通常的遍历操作
  - $O(n^2)$  双重循环
  - $O(2^n)$  递归复杂度
- 空间复杂度
  - $O(1)$  原地操作, 不开辟额外的内存辅助空间
  - $O(n)$  开辟线性辅助空间

存储结构

- 顺序结构
  - 数组
    - 连续空间
    - 随机访问  $O(1)$
    - 插入删除  $O(n)$
    - 查找快, 插入删除慢
- 链式结构
  - 单链表
    - 不支持随机访问, 需要通过遍历来访问节点
    - 插入和删除只需要移动指针, 时间复杂度为  $O(1)$
    - 每个节点需要额外的空间存储指针, 需要的内存比数组大
  - 双链表
    - 在单链表的基础上, 除头结点外, 每个节点增加一个存放指向前驱节点的指针
  - 循环链表
    - 尾节点指向头节点
  - 静态链表
    - 借助数组, 伴随指向后继节点的指针, 插入和删除不用挪动元素
  - 应用
    - 链表
    - 热门项目里用来替代平衡树

逻辑结构

- 线性结构
  - 栈
    - 先入后出, 基于顺序或者链式都可以实现
    - 实际应用
      - 浏览器的前进与后退
      - 编辑器中do/undo的实现
      - 括号匹配
      - 表达式计算
  - 队列
    - 普通队列
      - 先入先出, 基于顺序和链式都可以实现
      - 出入队  $O(1)$  的时间复杂度
    - 双端队列
      - 队列的出入口都可以进行出入队
    - 优先队列
      - 根据优先级出队
- 散列表
  - get和set都是  $O(1)$  的复杂度
  - 散列函数的应用
    - 安全加密
    - 唯一标志
    - 数据校验
    - 负载均衡
    - 数据分片
    - 分布式存储
- 堆
  - 大顶堆
  - 小顶堆
  - 应用
    - 找第k大的元素
- 树
  - 特点
    - 顺序和链式都可以实现
    - 遍历方式
      - 广度优先搜索
        - 前序遍历
      - 深度优先搜索
        - 中序遍历
        - 后续遍历
  - 二叉树
    - 完全二叉树
    - 满二叉树
    - 二叉搜索树
    - 平衡二叉搜索树
    - 红黑树
  - 哈希曼树
    - 最优二叉树
  - 字典树
    - Tribe树
    - 用于搜索引擎的词频统计
- 图
  - 存储
    - 顺序: 邻接矩阵
    - 链式: 邻接表
  - 拓扑排序
  - 最短路径
  - 关键路径
  - 最小生成树
  - 二分图
  - 最大流

特殊结构

- 并查集
  - 站队问题
  - 初始化
  - 查询、合并
  - 路径压缩
- LRU
  - 基于双向链表和散列表实现
  - get和set都是  $O(1)$  的复杂度
- 布隆过滤器
  - 判断不存在100%准确
  - 判断存在有误差
  - 利用hash函数将key分散在多个位置上

字符串匹配

- 朴素
- RK
- BM
- KMP
- Tribe
- AC自动机
- 后缀数组

查找

- 二分查找
  - 有序
  - 有界
  - 能够通过索引随机访问
- 线性查找
- 树查找
- 散列表查找

搜索

- DFS
- BFS
- A\*启发式

算法思想

- 动态规划
  - 简单版本利用递归+缓存
  - 高级版本利用递推公式
  - 一些实际场景需要套用模板
  - 状态的定义
  - 最优子结构
  - 状态转移方程
- 贪心算法
  - 判断能不能贪心
  - 弱化版的动态规划
  - 贪心空间
  - 终止状态
  - 本层处理
  - Drill Down
  - 本层状态清理
- 递归/分治
  - 回溯
    - 解决广义的搜索问题
    - 从一组可能的解中, 选择一个满足解的情况
  - 优化
    - 剪枝
  - 枚举
    - 枚举

排序

- $O(n^2)$ 
  - 冒泡排序
  - 插入排序
  - 选择排序
- $O(n \log n)$ 
  - 冒泡排序
  - 归并排序
- $O(n)$ 
  - 桶排序
  - 计数排序
  - 基数排序